N87-20066

A Unified Approach to Computer Analysis and Modeling of Spacecraft Environmental Interactions

I. Katz, M. J. Mandell, and J. J. Cassidy
S-CUBED, A Division of Maxwell Laboratories
P.O. Box 1620, La Jolla, California 92038

In the past decade we have developed a good understanding of many spacecraft/environment interaction processes. These include geosynchronous orbit charging, high voltage sheaths, ram/wake density variations, and certain surface processes. There are also many processes of which we are aware, but do not yet understand. Some of the outstanding questions include broadband noise in the ram, multiple ion streams, and electron heating. Advancing our knowledge is complicated by the fact that the various processes interact with one another.

In recent years, we have had successes in modeling some aspects of environmental interactions. This modeling has involved building our theoretical and phenomenological understandings into large, three dimensional computer codes. Each such code requires several man-years of theory, programming, and verification. The NASA Charging Analyzer Program (NASCAP) (Katz et al., 1977; Katz et al., 1979; Mandell et al., 1984) models spacecraft charging at geosynchronous orbit; NASCAP/LEO (Mandell et al., 1982) models large, high voltage spacecraft in low earth orbit; and POLAR (Cooke et al., 1985) models the charging of spacecraft due to auroral electrons. Comparisons with experimental data show good agreement, and we have consistently employed experimental results (both ground test and flight data) to help develop the computer models.

Despite these and other successes, there are serious weaknesses in the present approach to computer modeling. The most obvious problem is limited access to the various codes. Most researchers don't have handy a computer with NASCAP/LEO installed on it, and there is no easy way for them to get it.

What's more, even a researcher with access to all the computer models available from all sources will probably not make good use of them, if only because he doesn't have time to learn the different user interfaces required by each code. And even if he understood all the interfaces, he still couldn't use the codes unless he also had access to the half-dozen different computers for which they were designed.

Fortunately, there are several historical forces at work to make the job easier. Hardware is getting cheaper and faster at a breathtaking pace. Equally important, there are new software techniques and packages that routinely solve problems which, until recently, were impossibly difficult. There are several independently developed packages, such as PATRAN[1] which may

1 PATRAN is a registered trademark of PDA Engineering, Santa Ana, California.

PRECEDING PAGE BLANK NOT FILMED

be used to define general, three-dimensional objects in a form suitable for use by a finite element computer code. Presently, an effort is underway to make one interactions model, NASCAP/LEO, compatible with PATRAN objects.

Even more important, there are now operating systems that are 100% compatible across entire lines of computers from several different manufacturers. UNIX[2] is the best example. You can write a program in standard FORTRAN, and it will run, without a single change, on all of the popular workstation computers available. Also, it will give the same answer, with the same accuracy. This kind of dependable interchangeability of parts is as important to the computing community today as it was to the manufacturing community in 1800, when Eli Whitney "amazed government representatives by assembling guns from pieces chosen at random from piles of parts." (Latham, 1967)

The experimentalists and the engineering community have already realized the benefits of these advances. They routinely use standardized instrument controllers, connectors, and data handling protocols. This allows them to focus their efforts on the unique scientific and developmental aspects of their particular experiments.

Characteristics of UNISIM

As a way to make use of all these advances, we propose a new, coordinated, unified approach to the development of spacecraft plasma interaction models. The objective is to eliminate the unnecessary duplicative work in order to allow researchers to concentrate on the scientific aspects. By streamlining the developmental process, we can enhance the interchange between theorists and experimentalists and speed the transfer of technology to the spacecraft engineering community. We call this approach the UNIfied Spacecraft Interaction Model (UNISIM).

UNISIM is a coordinated system of software, hardware, and specifications. It is a tool for modeling and analyzing spacecraft interactions. It will be used to design experiments, to interpret results of experiments, and to aid in future spacecraft design. It breaks a Spacecraft Interaction analysis into several modules. Each module will perform an analysis for some physical process, using phenomenology and algorithms which are well documented and have been subject to review. The result is a system with the following features:

- Modularized software (object oriented);

- Generalized geometry;

- Peer review for new modules;

- Open system, coordinated effort;

---

2 UNIX is a trademark of Bell Laboratories, Murray Hill, New Jersey.

- Codes, documentation, and information exchange via network;

- Artificial Intelligence based user interface;

- Standardized coding, documentation, and units.

Some of these concepts are already in use for other scientific areas, such as the NASA/Langley IDEAS package (Integrated Design and Evaluation of Advanced Spacecraft; Garrett, 1981; Wright et al., 1984). Tying together available geometric and interactions models via a commercially available CAD/CAM data base has been suggested by P. R. Williamson (Private Communication). The uniqueness of UNISIM centers on the openness of the system and the focus on coordinated multiresearcher participation along with the scientific peer review process. The idea underlying UNISIM is to extend scientific communication from just the print medium to both print and electronic media for the field of spacecraft interactions modeling.

Modularized Software

The current method of developing computer models requires a lot of redundant effort. Every individual computer code includes certain sections - object definition, grid generation, matrix calculation, graphical results display, and so on - each of which is reproduced in a slightly different form in every other code. The writing of these essentially similar parts often takes more effort than working out the details of the science.

Figure 1 illustrates the type of structure we envision for UNISIM. Each task of the sort mentioned in the preceding paragraph would be handled by a utility module which could be used for any scientific purpose. Then, if you want to add a new scientific model, you only have to create a single new module to plug into the overall system.

Not only will scientific modules be dramatically easier to write than currently, but each module will have easy access to calculations by other modules. For example, a high voltage collection calculation needs to know the variations of the plasma density caused by wakes. This kind of data availability is an important feature of UNISIM.

One important aspect of these modules is that they communicate by sending requests for information back and forth. One module does not have to know how the other module does its work; it just needs to know how to obtain the data calculated by the other module, or how to request a calculation if the data is not current. In this way, true independence between modules is maintained. This is a concept of Object Oriented Programming (Love, 1983; Ledbetter and Cox, 1985), which we will implement to the degree possible within the programming language used.

Generalized Geometries

An experiment on a space station with other experiments around it is inherently a three dimensional problem. The spacecraft itself is not symmetrical, and, commonly, other influences such as sunlight or variations in the environment destroy any symmetries that may exist. To get results that are usable in the real world, it is necessary that computer codes allow general, three dimensional spacecraft models.

UNISIM will accept geometric input from commercial CAD/CAM solid modelers to allow accurate specification of the spacecraft and experiment geometries. Many modules will include subgrid refinement to resolve small details of instruments.

Peer Review Process

UNISIM will include a formal peer review process for the addition of new modules to the system. This will include analysis of the scientific approximations employed in the model, verification of the algorithms, and verification that the code actually executes the algorithms correctly.

Peer review has long been a requirement for scientific papers published in the open literature. By extending this practice to computer codes, we ensure that users can have confidence in the results they get using UNISIM.

Open System, Coordinated Effort

The UNISIM specifications and requirements will be openly available, allowing space scientists at various sites to contribute. All users will be able to communicate with each other about results, problems, and suggestions for further work.

Module specifications will be explicitly stated, so anyone will be able to design a new module, or substitute at his own site a locally written module in place of the one normally used.

To make it clear what each module does, source code and a complete description of the algorithms used will be included in the module itself. Then any change in a module will be accompanied by a simultaneous change in the on-line documentation.

The openness of the system makes it easy to add new features as our understanding of the physics advances. New modules can be introduced with ease, and old modules can be superseded or replaced without disrupting the system.

Network Availability

One of the fundamental features of UNISIM is that it will be available over a network such as SPAN or ARPANET. This means that as soon as a new module is included, it will be available to everyone. The developers will not be plagued with requests for installation, the users will not have to wait impatiently for updated capabilities, and no one will have to fiddle with magnetic tapes.

In addition to being a medium for access to the program itself, the network will provide an information exchange including a catalog and descriptions of the online modules. Through a bulletin board or an electronic newsletter, bugs, errors, and other problems can be identified and corrected quickly.

User Interface

Scientific codes which model spacecraft interactions are complicated to use due to their highly technical nature. It requires a certain level of expertise just to know what a program is supposed to do, and what kind of input is meaningful. On top of this, the user needs to know the instructions and commands that the program accepts.

One of the great benefits of a modular system like UNISIM is that a single user interface can be used for the entire system. This saves the user from needing to learn a multiplicity of interfaces, and it allows the development effort to focus on making a single natural, easy to use interface. In particular, it makes it profitable to take the time to apply Artificial Intelligence principles, such as expert system techniques, to making the modules usable by scientists and engineers who are not specialists in computer simulation.

Standards

The concept of standards is extremely important to UNISIM. The system requires standards in three areas - coding, documentation and units. All programming will be done in ANSI Standard FORTRAN. The use of a standard operating system, preferably UNIX, assures transportability of both source code and data files. (UNIX is not tied to a single computer manufacturer. Other operating systems are also usable, but some of the transportability is lost.) A precision standard (e. g. IEEE floating point) will ensure that all modules will produce same results regardless of which machine is used to actually run the code.

Standard programming techniques will be enforced. These will include such things as a standard form for control structures, required comment headings for all subroutines, and other internal documentation standards. Data access will be through separately compiled subroutines, so that the individual data and file structures are hidden. Variable names will be as descriptive as possible, in order to enhance the readability of the source code. The use of a uniform coding style will not only help researchers who want to look at the details of the algorithms, but it will be a great help in maintaining and debugging the codes.

User manuals for each of the modules will also conform to appropriate standards. This task will be simplified because all modules will use the same utility modules for tasks like input and output, which are often the most confusing part of a manual. Manuals will contain physical models, algorithms, numerical techniques, and program, file and data structures.

All computational results will be specified in Systeme International units (Mechtly, 1973). Unit confusion is often a big problem in codes that report their results in "code units" or in non-dimensional units. But even if a module uses unique units for internal computation, it will use standard units to interact with the other modules. Standard "include" files with names and values for common physical quantities will be available to the individual module developers.

IMPLEMENTING UNISIM

The first step to implementation of UNISIM is to create an overall system definition. This will include a definition of what constitutes a module and how modules communicate with each other. The module definition will be general enough to contain currently planned modules and new modules which cannot yet be foreseen. It will include specifications both for the code parts and for data protocols for communication between modules. Also required will be the procedures for developing and accessing experimental modules, for the peer review process, for including modules which have passed the review process, and for revising or deleting previously qualified modules.

The first modules to be built will be the utility modules. Since the PATRAN solid object modeler is used at several NASA centers, a PATRAN to UNISIM translator will be a good candidate for early inclusion. This would be a natural extension of the present conversion of NASCAP/LEO to use PATRAN objects, but there are now a number of CAD-based solid object modelers. UNISIM will not contain any commercial, proprietary modules.

All the concepts in the world of generalized geometry don't help if you don't have algorithms that can use them to do the calculations. We are currently developing a Generalized Blended Element algorithm which is for implementation in NASCAP/LEO. This is a way to automatically generate matrix elements for Poisson's equation (or any elliptical equation on a three-dimensional grid) in the space surrounding a general object. The method is quite general, allowing grid elements of any shape, with any desired resolution. The algorithm generates the matrix elements completely automatically.

Spacecraft interaction calculations that work with real spacecraft models also require sophisticated graphical output. There are now several packages and terminals which are specifically designed to display three dimensional models and calculational results. Figure 2 shows wireframe, surface material, and surface potential plots of a spacecraft defined using NASCAP as a CAD program. Figure 3 shows a surface material plot of a spacecraft defined using SLIC[3] as a CAD program.

UNISIM graphics utilities will be designed to be easily interfaced to all such facilities. It will also be possible to transfer and exchange data via standard file structures such as IGES (Smith and Wellington, 1986) or MOVIE.BYU[4] (Christiansen and Stephenson, 1986).

---

3 SLIC is a product of GCN/Hydronet Services, Stockton, CA.

4 MOVIE.BYU is distributed by Graphics Utah Style, 1980 North 1450 East, Provo, Utah 84604.

BENEFITS OF UNISIM

UNISIM offers benefits at every step of its implementation. It advances the technology of spacecraft environment interactions modeling. Each of its features is something that has to be done anyway, and implementing the whole thing in a coordinated effort greatly decreases the costs in time and money.

UNISIM will directly cause a closer coordination between theory, experiment, and engineering. The instant availability of the UNISIM codes will allow experimentalists to get computational results without waiting weeks or months. Theorists will benefit greatly from a freer exchange, both among themselves and with the experimentalists, caused by the open nature of the system.

UNISIM supports Telescience (Black, 1986) by making available computer models of spacecraft environment interactions for use during mission simulation and planning. The same geometrical descriptions that are used for mechanical simulation and analysis can be used for environment interactions modeling.

UNISIM enables scientific issues to be addressed more quickly because the theorist can focus on the physics rather than the rest of the required coding. Since computations and comparisons with experiment can be performed and reported more quickly, weaknesses in the theories will be exposed sooner, leading to improved theories.

UNISIM makes transfer of the developed technology to spacecraft designers and engineers simple, because the whole approach is compatible with the existing engineering software which they already use. Structural analysis is routinely performed using solid modelers, finite element mesh generators and analysis packages with the results shipped to a graphics display device.

UNISIM increases the coordination and reduces the risk of developing spacecraft environment interactions models as transferable technology. Since all the models are part of a single, well documented, system, managing the development and communicating the advances among the space science community is relatively simple. And, finally, the risk associated with the system is small, since there are no irreplaceable links, no critical pieces; rather, there is a prescription for generating pieces that will work together.

ACKNOWLEDGEMENT

The authors have benefitted from discussions of this concept with William N. Hall.

REFERENCES

Black, D. C., "Telescience in Orbit", Aerospace America, September 1986, pp. 44-46.

Christiansen, H. and M. Stephenson, MOVIE.BYU Training Text, Community Press, Provo, Utah, 1986.

Cooke, D. L., I. Katz, M. J. Mandell, J. R. Lilley, and A. Rubin, "Three-Dimensional Calculation of Shuttle Charging in Polar Orbit", Spacecraft Environmental Interactions Technology 1983, NASA CP-2359, AFGL-TR-85-0018, 1985.

Garrett, L. B., "Interactive Design and Analysis of Future Large Spacecraft Concepts", AIAA Paper 81-1177, June 1981.

Katz, I., D. E. Parks, M. J. Mandell, J. M. Harvey, D. H. Brownell, S. S. Wang, M. Rotenberg, A Three Dimensional Dynamic Study of Electrostatic Charging in Materials, NASA CR-135256, S-CUBED Report SSS-R-77-3367, 1977.

Katz, I., J. J. Cassidy, M. J. Mandell, G. W. Schnuelle, P. G. Steen, and J. C. Roche, "The Capabilities of the NASA Charging Analyzer Program", Spacecraft Charging Technology - 1978, NASA CP-2071, AFGL-TR-79-0082, 1979.

Latham, J. L., "Whitney, Eli", The World Book Encyclopedia, 1967 edition, vol. 20, p. 247.

Ledbetter, L. and B. Cox, "Software-ICs: A plan for building reusable software components", BYTE, June 1985, p. 307.

Love, T., "Application Development in Small-Talk-80", Arlington VA: Proceedings of Softfair-83.

Mandell, M. J., I. Katz, and D. L. Cooke, "Potentials on Large Spacecraft in LEO", IEEE Trans. Nucl. Sci. NS-29, pp. 1584-1588, 1982.

Mandell, M. J., P. R. Stannard, and I. Katz, NASCAP Programmer's Reference Manual S-CUBED Report SSS-84-6638, 1984.

Mechtly, E. A., "The International System of Units, Physical Constants, and Conversion Factors,", 2nd Revision, NASA SP-7012, 1973.

Smith, B. and J. Wellington, Initial Graphics Exchange Specification (IGES), Version 3.0, U. S. Department of Commerce NBSIR 86-3359, Gaithersburg, MD, April 1986.

Wright, R. L., D. D. DeRyder and M. J. Ferebee, "Interactive Systems Design and Synthesis of Future Spacecraft Concepts", NASA TM-86254, Langley Research Center, June, 1984.

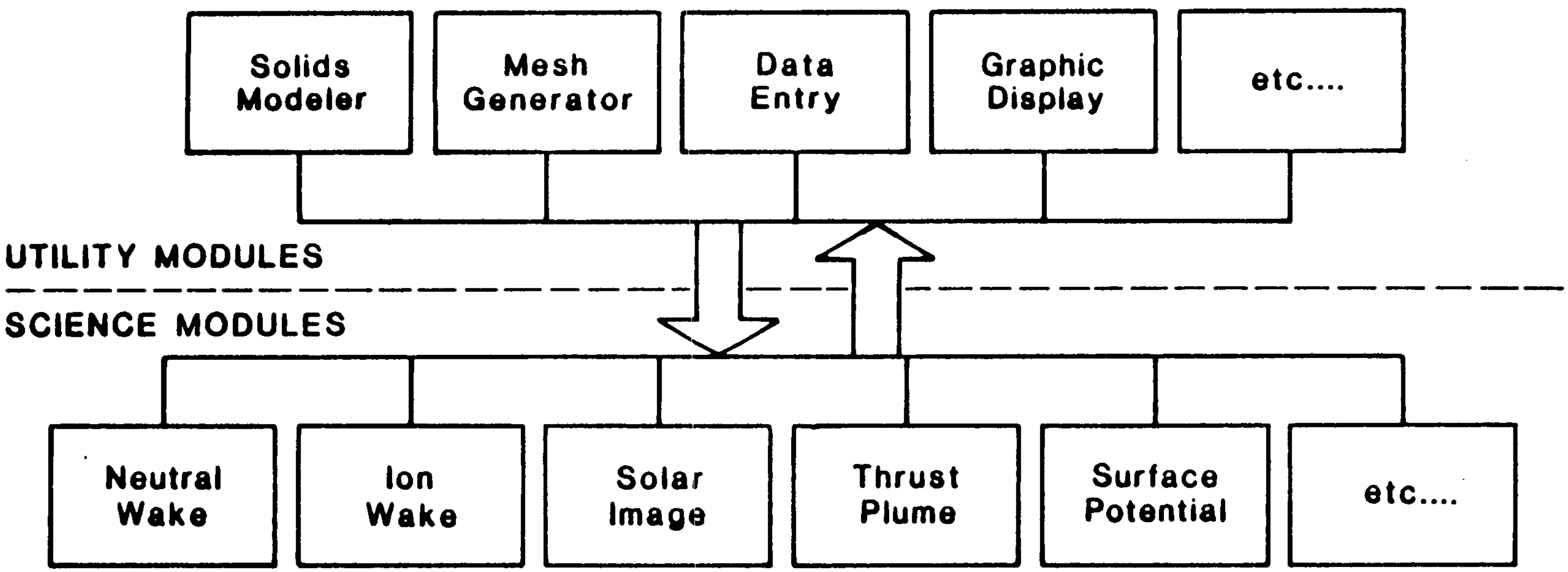

Fig. 1. Block diagram showing the modular structure of UNISIM.

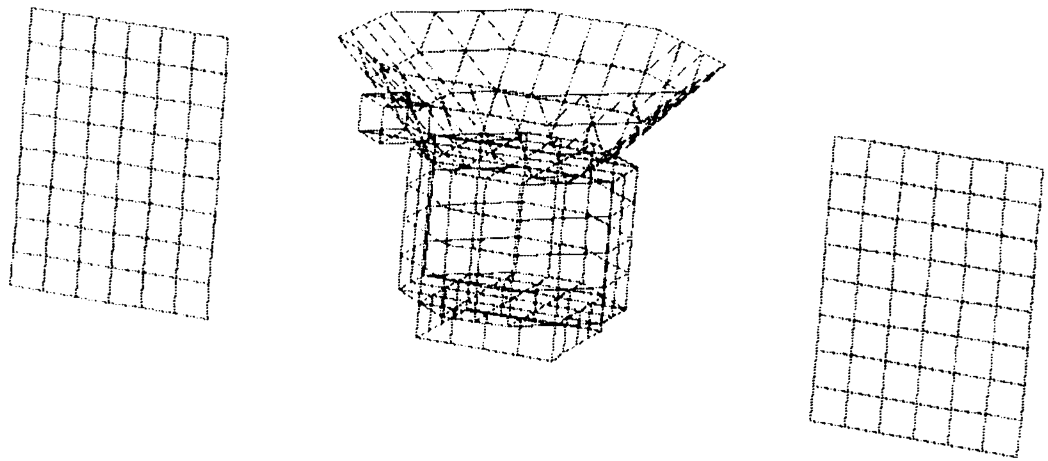

Fig. 2a. Wire frame model of spacecraft defined using NASCAP as a CAD program.

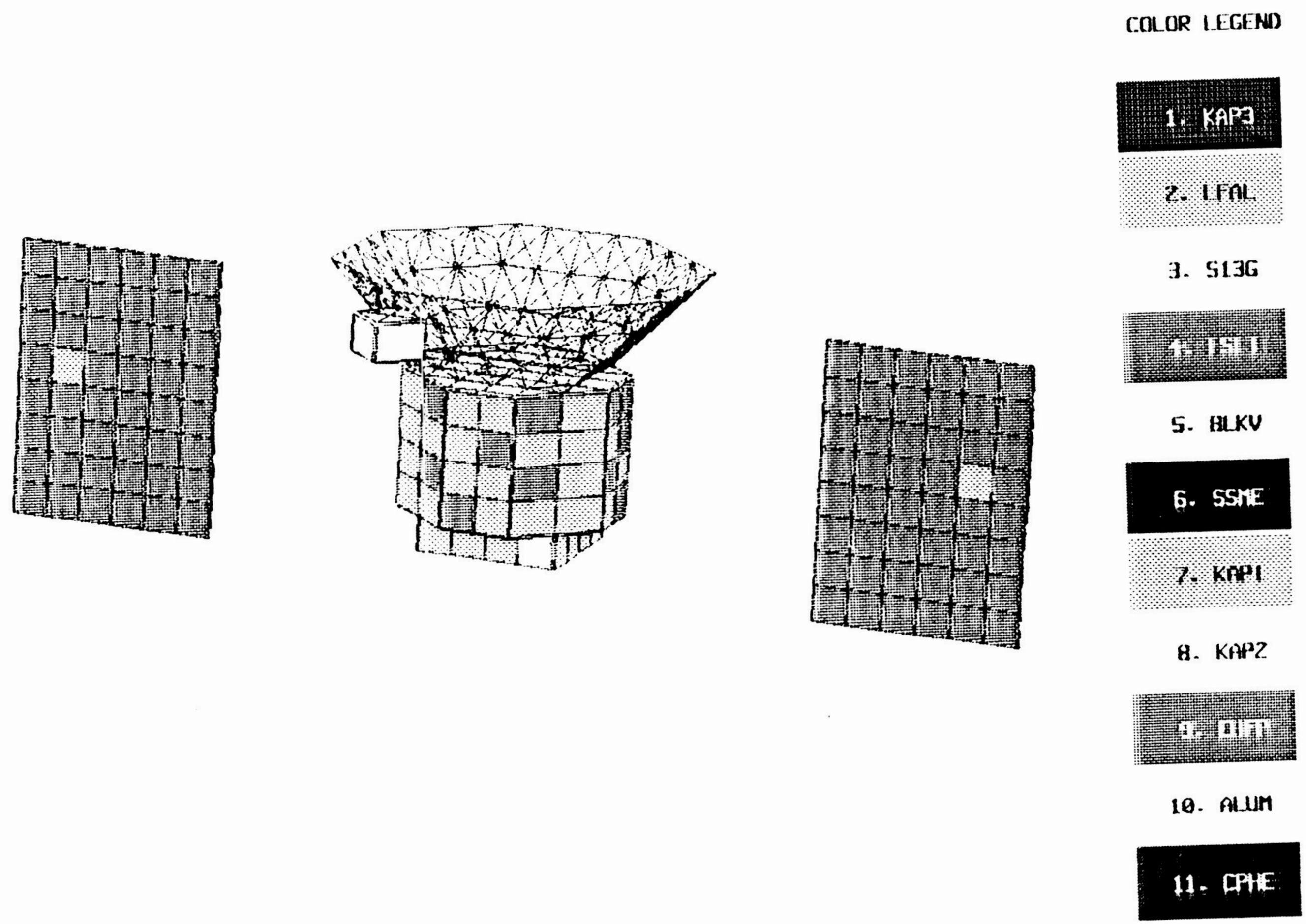

Fig. 2b. Surface material plot of the same object as figure 2a, with orientation preserved. The same plotting algorithm could be used for any discrete-valued property. [The original colors have been replaced by grays to minimize reproduction costs.]

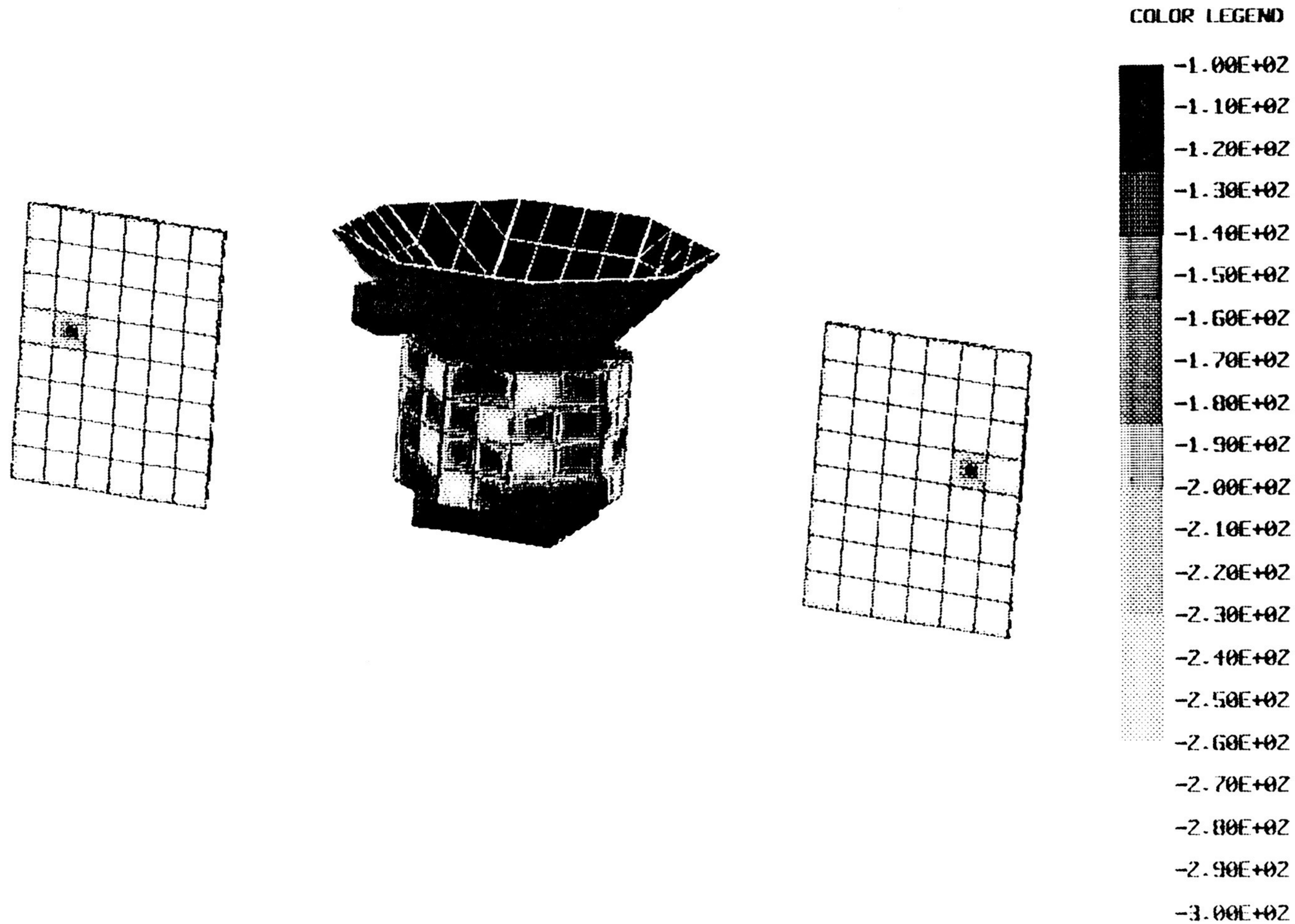

Fig. 2c. Surface potential plot of the same object as figure 2a, with orientation preserved. The same plotting algorithm could be used for any continuous-valued property. [The original colors have been replaced by grays to minimize reproduction costs.]

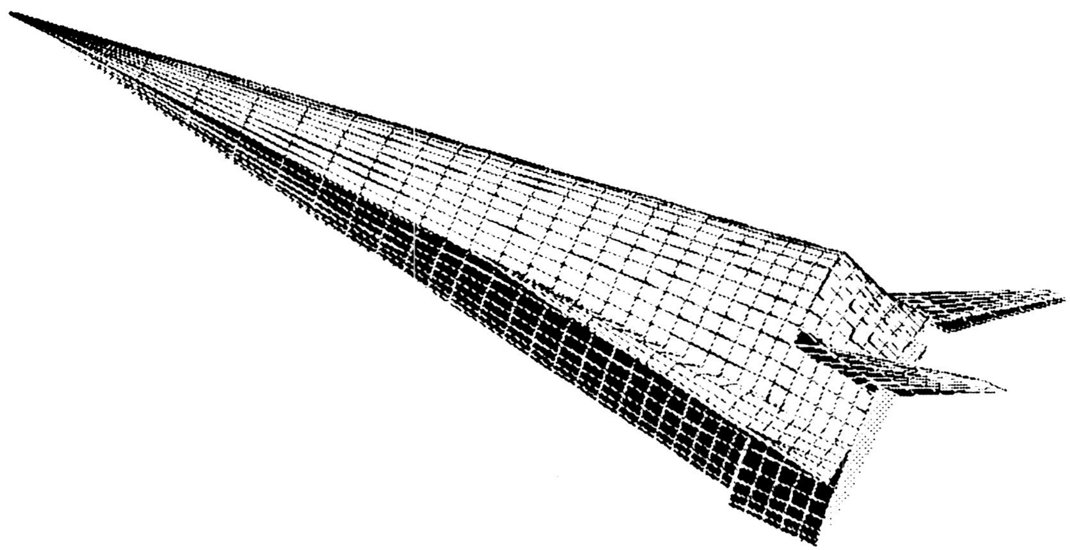

Fig. 3. Surface material plot of an object defined by the SLIC program. [The original colors have been replaced by grays to minimize reproduction costs.]